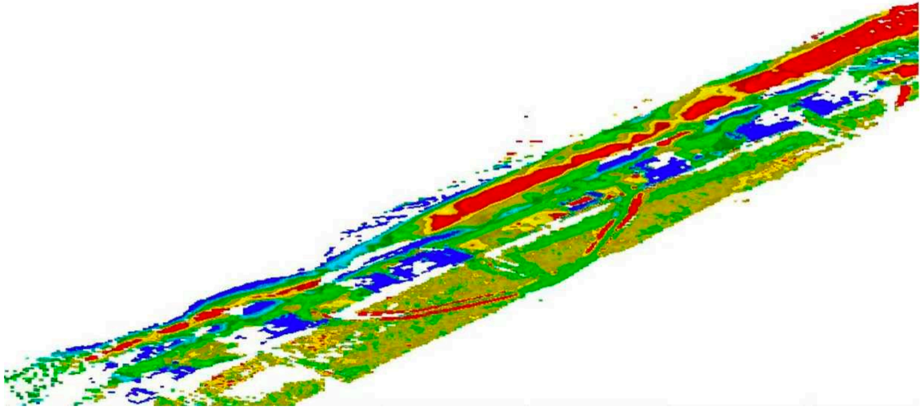
Coastal change from 4D point cloud data

1) NWO TTW – CoastScan, 2018 – 2023, website: https://coastscan.citg.tudelft.nl/

2) NWO TTW – AdaptCoast, 2024 – 2027

Kijkduin beach difference (2017-04-02)

Compared to 2016-11-11. Color: Red=+0.5 m - Blue=-0.5m

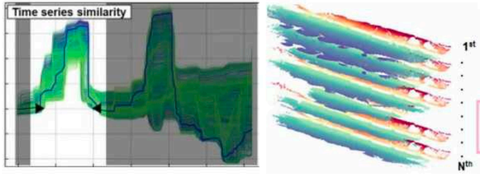


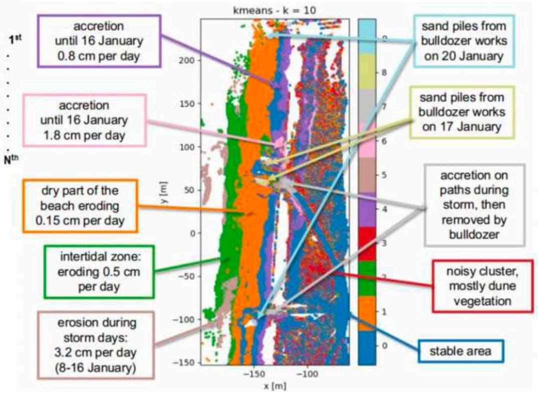
Motivation: Dutch dune & coastal dynamics

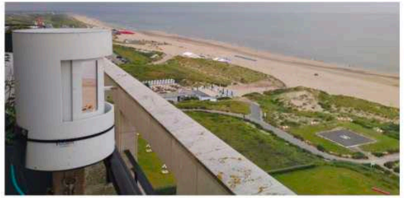
https://dezandmotor.nl/en

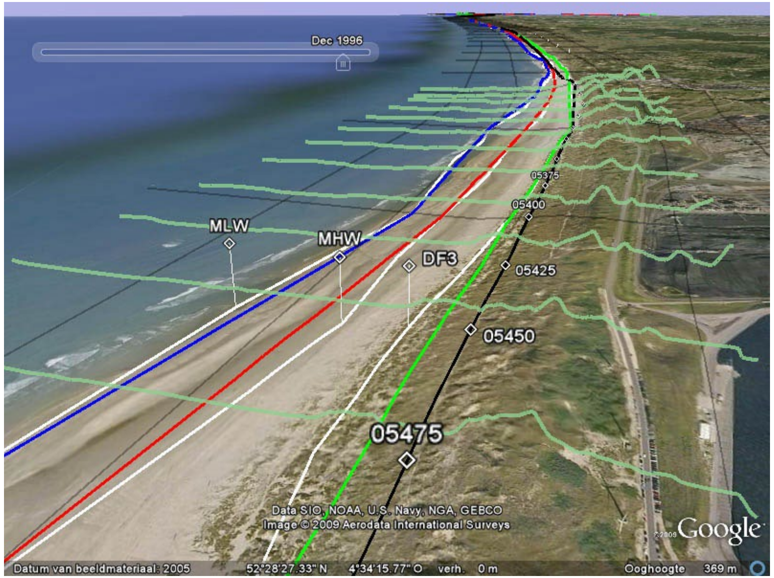
Coastal 4D point clouds

Permanent laser scan data ->
10000+ hourly epochs









<- Yearly Coastal (LiDAR) measurements
10+ yearly epochs

Snippet:
http://doris.tudelft.nl/~rlindenbergh/vgc/

Change detection

1) Consider two 3D data sets of a given scene:

   – Before

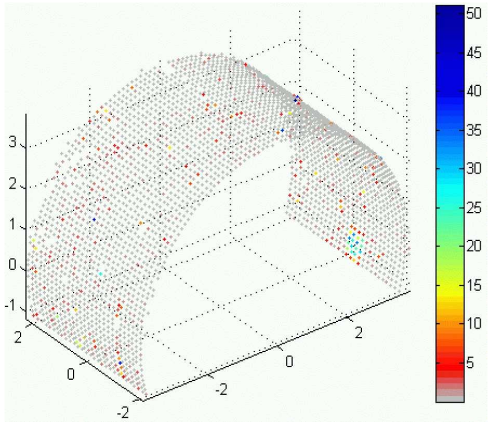
   – After

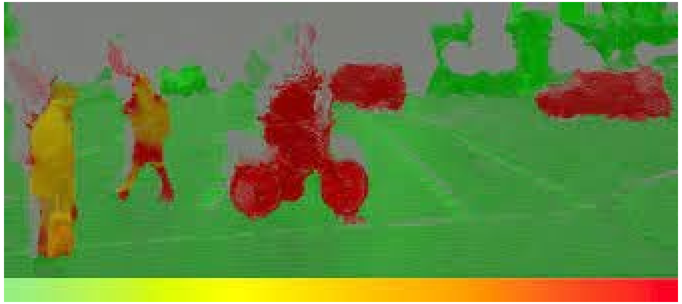
2) Did the scene change?

3) How confident are we that the scene changed, given the quality of the data?

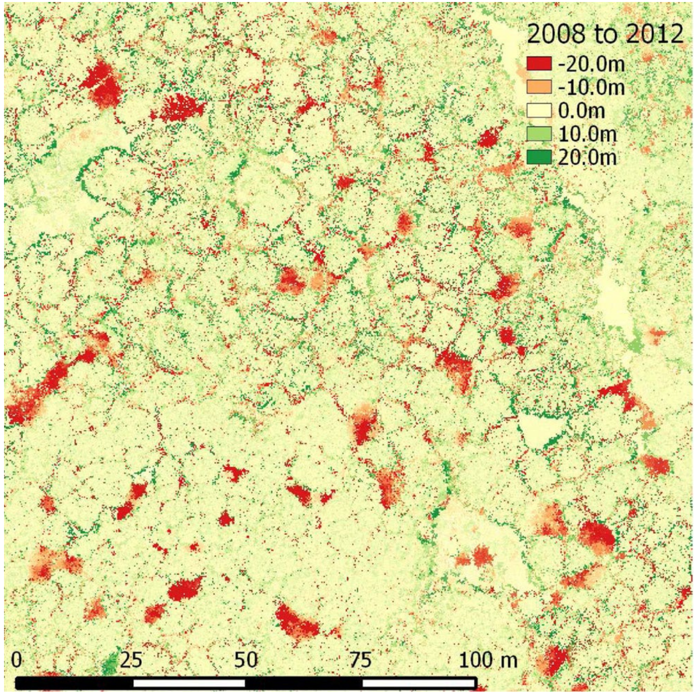
4) How much did the scene change?

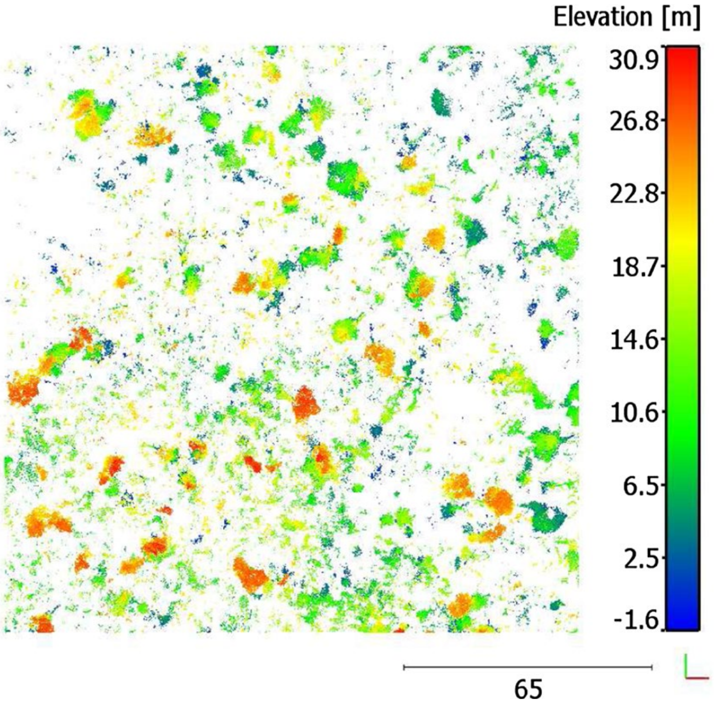
5) In which direction did the scene changed?





2D vs. 3D change

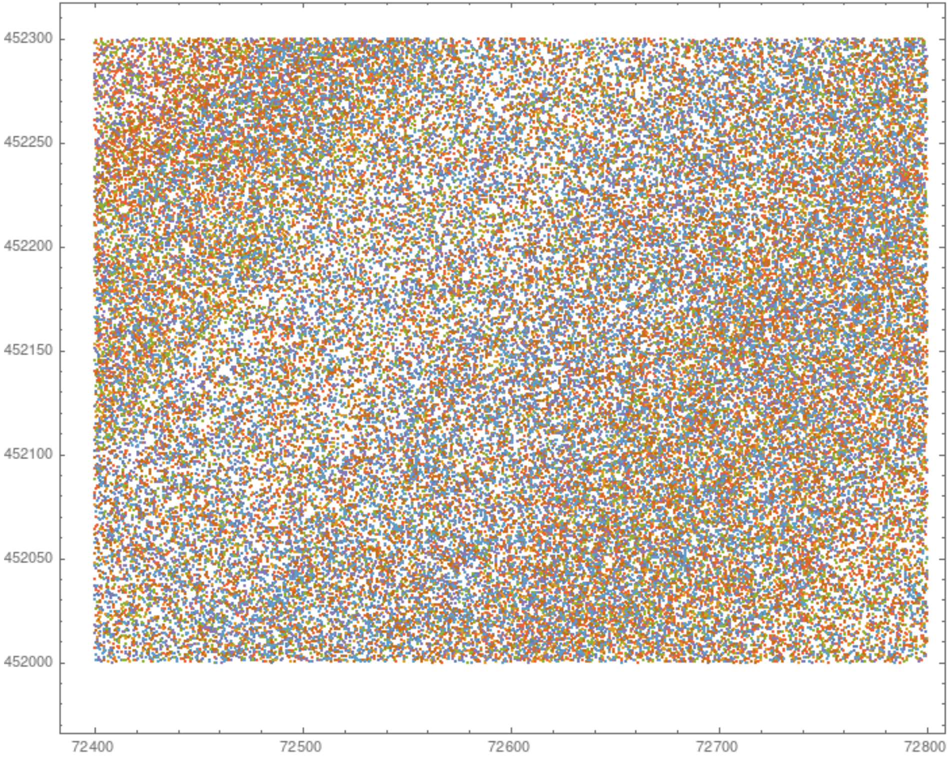




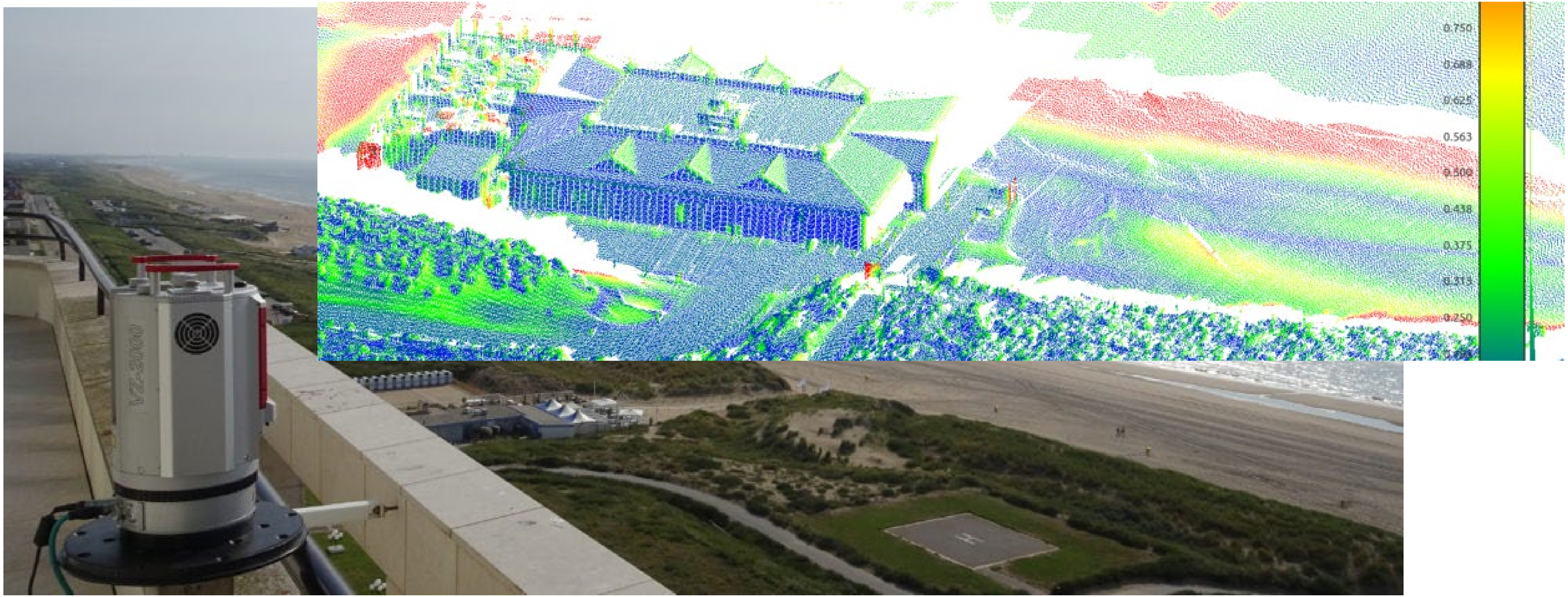
Challenges

Data issues:

- Point clouds are not aligned
- Different epochs, or even different points in point cloud, have different (unknown) quality
- Point clouds are too big
- Points from different epochs are at different locations
- ...

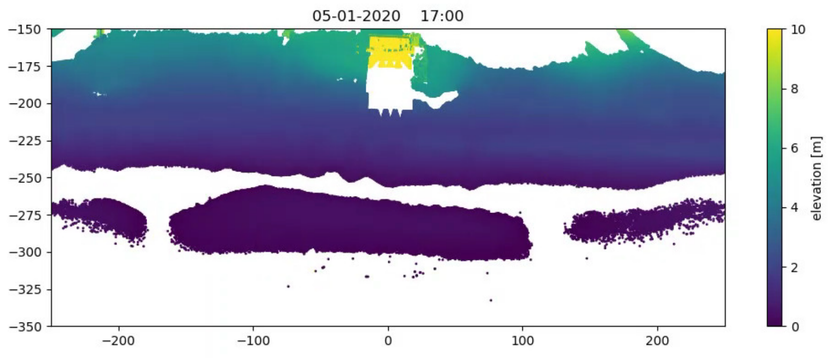


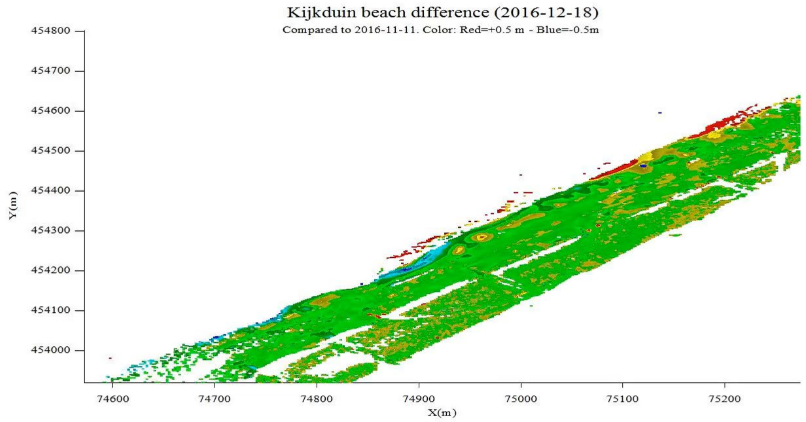
Automatized, repeated acquisition



Vos, S., Anders, K., Kuschnerus, M., Lindenbergh, R., Höfle, B., Aarninkhof, S., & de Vries, S. (2022). A high-resolution 4D terrestrial laser scan dataset of the Kijkduin beach-dune system, The Netherlands. *Scientific Data*, *9*(1), 191.

4D Data





Videos: https://coastscan.citg.tudelft.nl/index.php/data-publication/

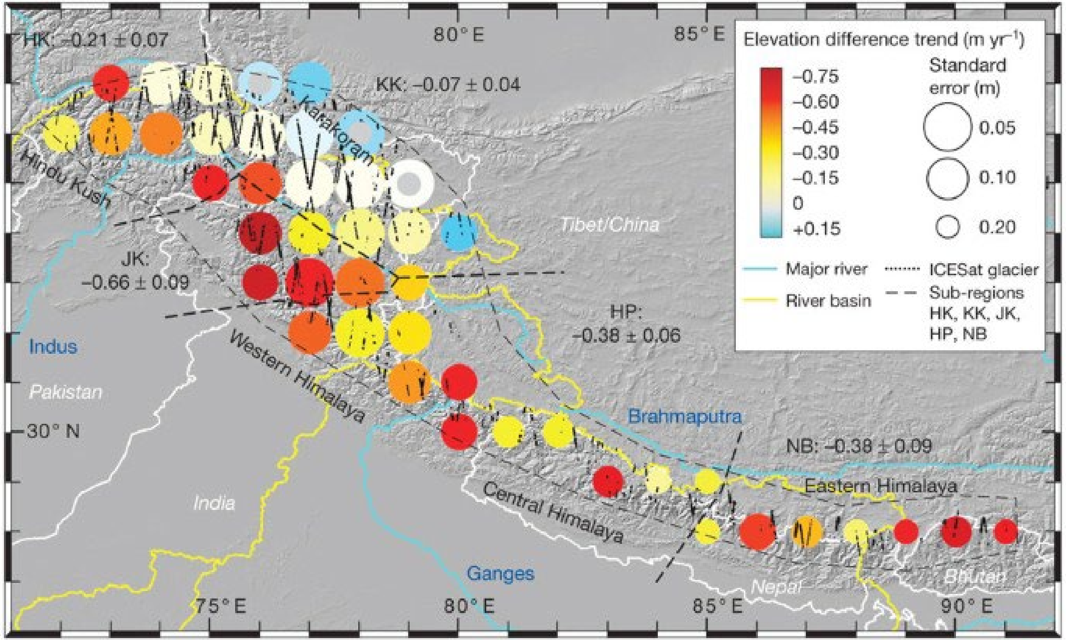
4D Data: repeatedly available 3D point cloud data with a significant temporal dimension

Methods for 4D analysis

Consecutive difference with first epoch/reference DEM

Kääb, A., Berthier, E., Nuth, C. *et al.* Contrasting patterns of early twenty-first-century glacier mass change in the Himalayas. *Nature* **488**, 495–498 (2012). https://doi.org/10.1038/nature11324

Applied on non-repeated ICESat profiles



Methods for 4D analysis

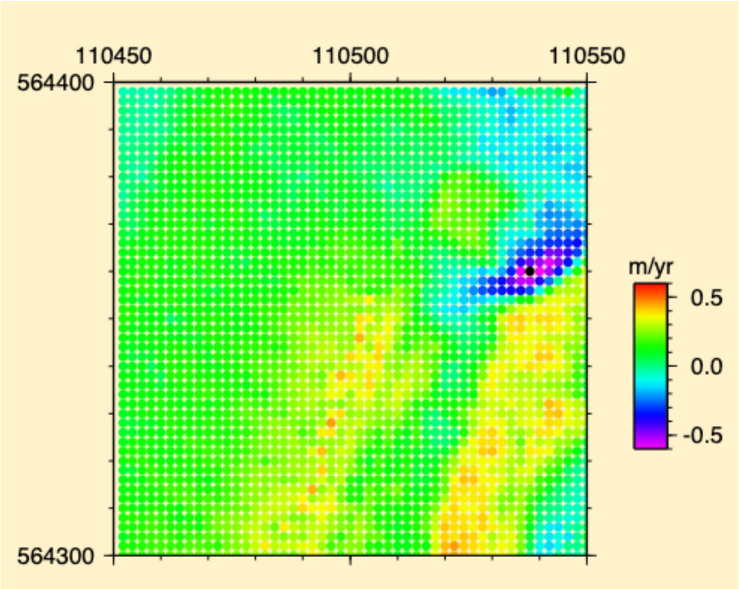
Trend Analysis
- http://doris.tudelft.nl/~rlindenbergh/publications/igarssposter.pdf
- Kalman filtering, https://doi.org/10.5194/esurf-11-593-2023
- Advanced version, under development, Mieke Kuschnerus

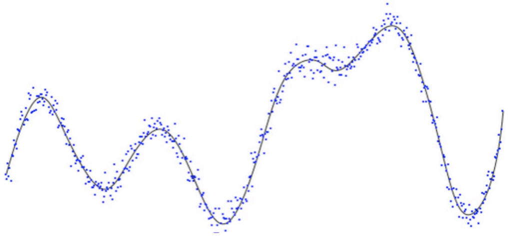
Coefficients of fitted functions
- Reduces data volume
- Allows to interpolate missing epochs

Clustering@Time Series: Mieke Kuschnerus

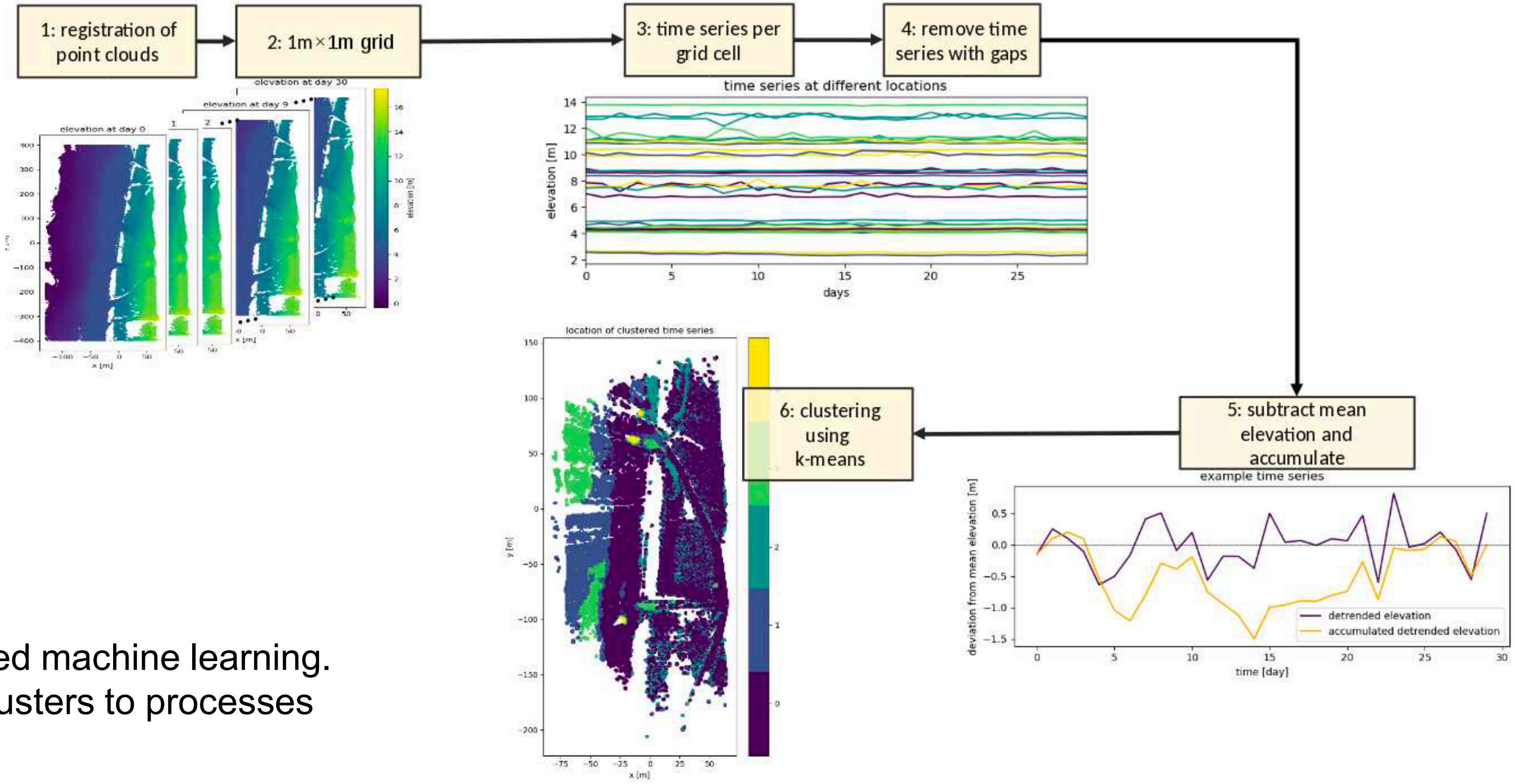
PCA@Time Series: Paco Frantzen

4D objects-by-change: Katherina Anders & Daan Hulskemper





Clustering time series of PLS data



Unsupervised machine learning.
Next: link clusters to processes

Principal Component Analysis

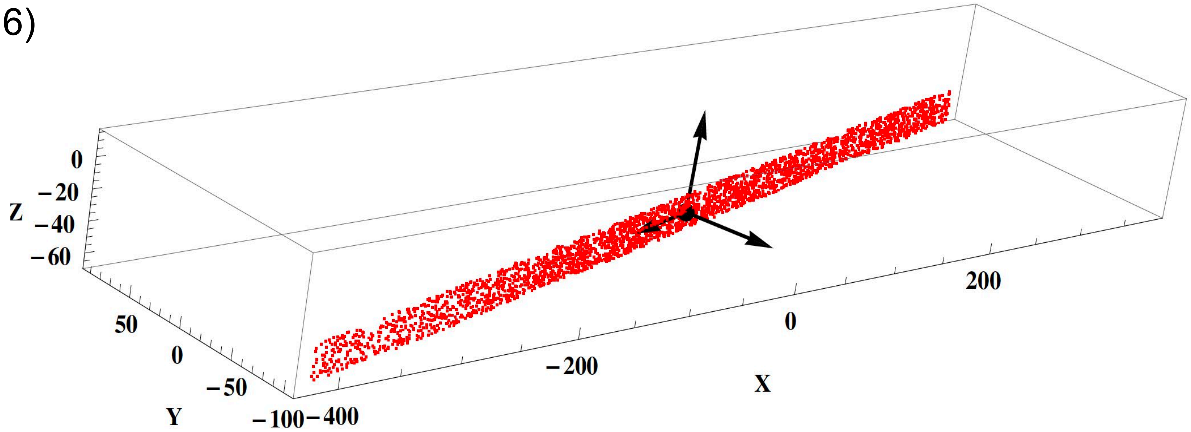
PCA: classic method for multivariate data analysis and remote sensing:

Python notebook for analysing multi-spectral satellite data using PCA:

https://docs.digitalearthafrica.org/en/latest/sandbox/notebooks/Frequently_used_code/Principal_component_analysis.html

PCA for Point Clouds

T. Hackel, J. Wegner, K. Schindler, Contour detection in Unstructured 3D Point Clouds, IEEE CVPR, (2016)

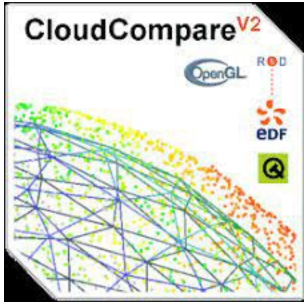


Key idea for 3D shape analysis:
determine consecutive, othogonal directions of maximal variability

=> 3 directions for 3D

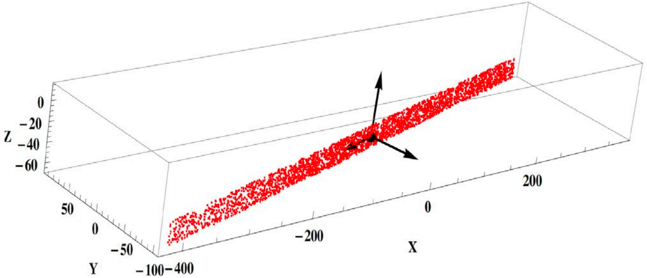
Question: how many directions for nD?
Question: how many dimensions will we have?

How to use PCA for point clouds?



1. Input: (k x 3) data matrix of k 3D points
   P
2. Determine the matrix of (k x the same) 3D mean/center of the k points:
   M_P
3. Get the centralized version of P:
   $P_c = P - M_p$ (still size k x 3)
4. Determine the data covariance matrix
   $C_P = (1/k)\, P^T_C \cdot P_C$ (size 3 x 3, T stands for transpose)
5. Output: three eigenvectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ and corresponding ordered & positive eigenvalues λ_1, λ_2, λ_3 of C_P

- Eigenvectors point in (remaining) max variability directions
- Eigenvalues provide size of these variability



CloudCompare: eigenvalues based shape features
(all kind of combinations of λ_1, λ_2, λ_3)

PCA loadings for point clouds

Our 3 3D eigenvectors, $\underline{\mathbf{e}}_1$, $\underline{\mathbf{e}}_2$, $\underline{\mathbf{e}}_3$, provide an alternative Cartesian coordinate system of our 3D space.

For each individual point $p \in P$, one can determine its PCA coordinates

Definition: Loadings of $p \in P$: PCA coordinates/coefficients of p

Obtaining the loadings L of P:

$$L = P_C \, . \, E$$

With E the matrix with the Eigenvectors as columns

(L consists basically of the coefficients of the points in the new PCA basis)

PCA for 4D time series

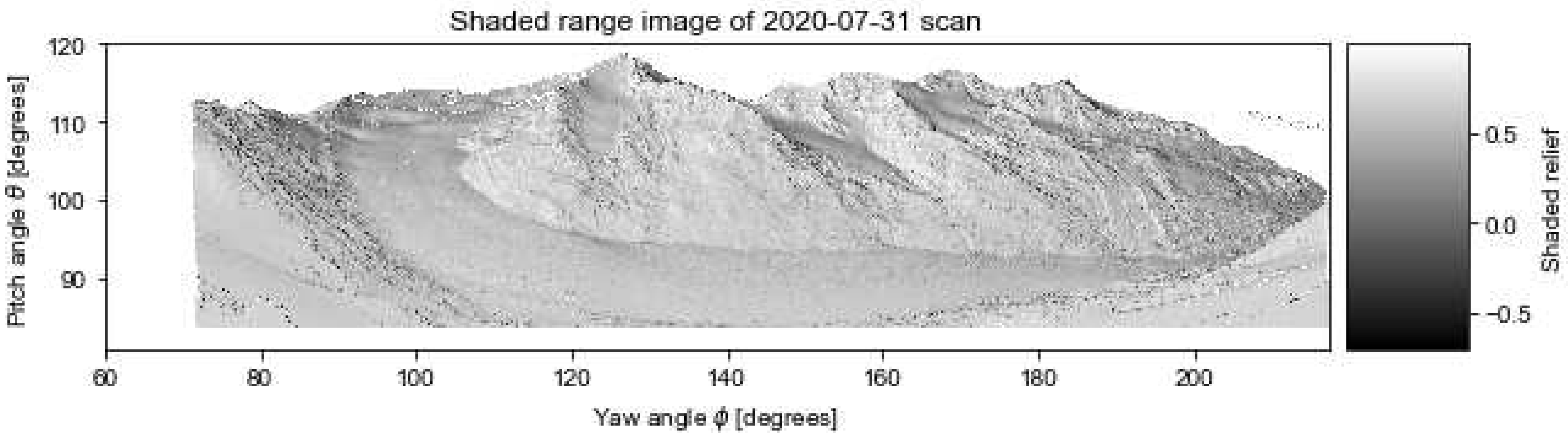
Paco Frantzen,

Identifying high Alpine geomorphological processes using permanent laser scanning time series,

MSc thesis, TU Delft (research done at Univ. Innsbruck)
Journal version: under review

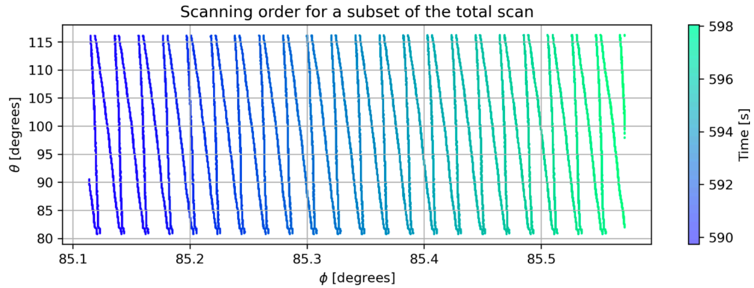
http://resolver.tudelft.nl/uuid:ce98c4e3-6ca1-4966-a5cf-2120f2fa44bf

Hintereisferner, Austria

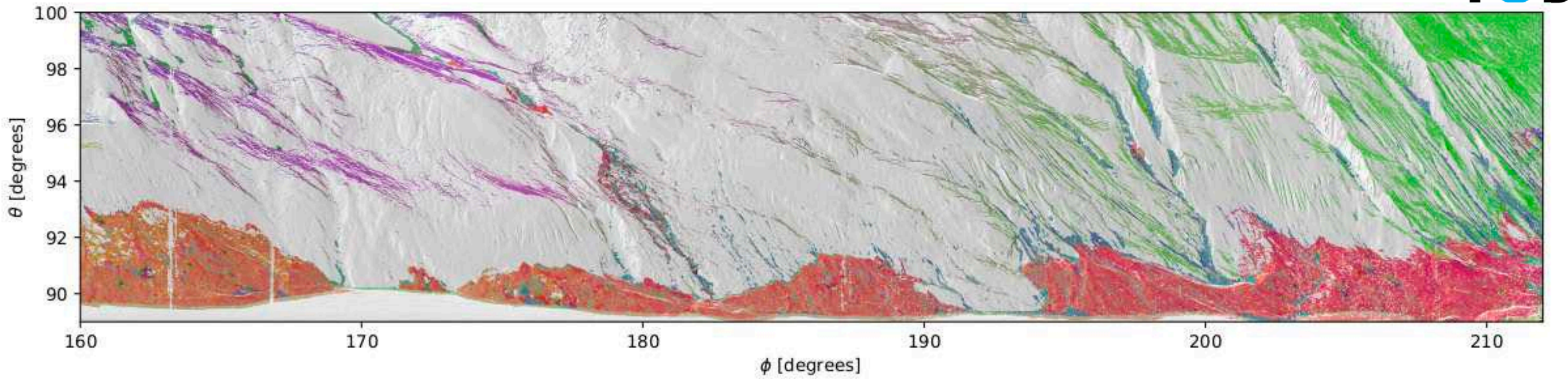


To analyze morphological change above the Hintereisferner, Paco Frantzen used

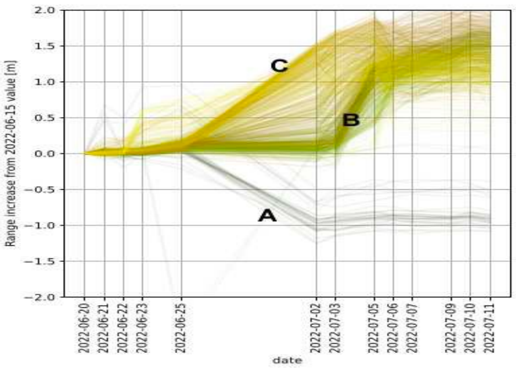
- Range images, the native scanner format
- Space time array, list of time series of deviations per range cell
- PCA: highlight principal deviation patterns

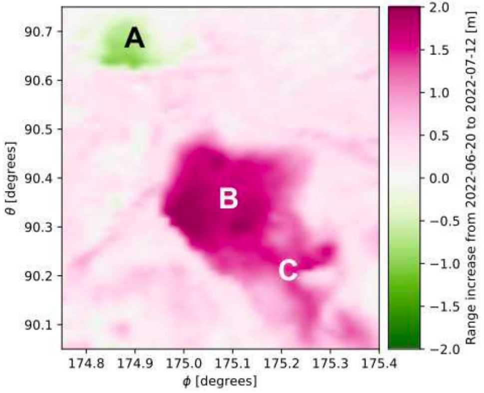


Geomorph-analysis: PCA@time series



False colour image of relative loadings per raster cell for the first three (combined) PCs of 2022 data. Red, green, and blue correspond to the first, second, and 3 and 4 combined PC, respectively.





PCA for 4D point clouds

Assume: consecutive point clouds represent change in elevation

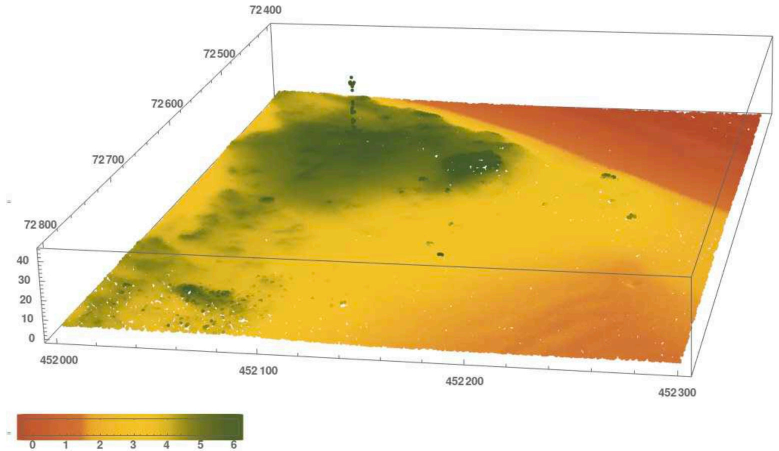
Idea: look for patterns in the elevation deviations.

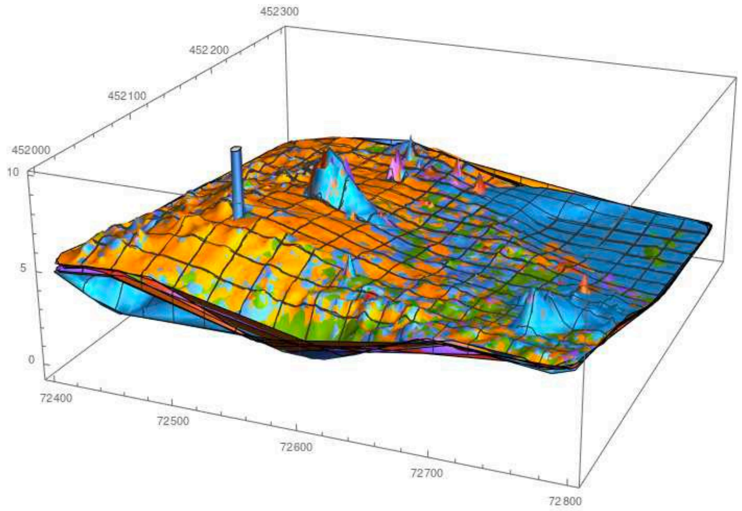
1. Input: list of k time series of elevations at m moments:
   $((x(1),y(1)), (z_1(1), z_2(1) \ldots, z_m(1)))$
   …
   $((x(k),y(k)), (z_1(k), z_2(k) \ldots, z_m(k)))$
2. Select a reference elevation, e.g. the elevation at m=1 and covert to deviations from this elevation:
   $((x(1),y(1)), (0, \Delta z_2(1) \ldots, \Delta z_m(1)))$
   …
   $((x(k),y(k)), (0, \Delta z_2(k) \ldots, \Delta z_m(k)))$
   With $\Delta z_i = z_i - z_1$

Now apply PCA to find variability in this (m-1) dimensional ‘deviation space‘

(forget the locations (x,y))

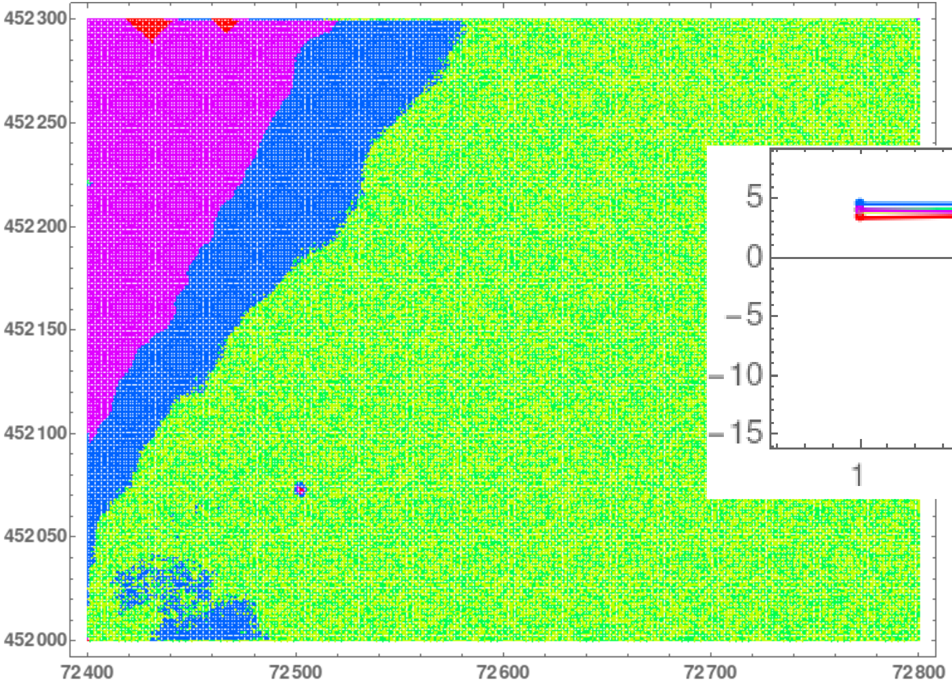
Input topographies

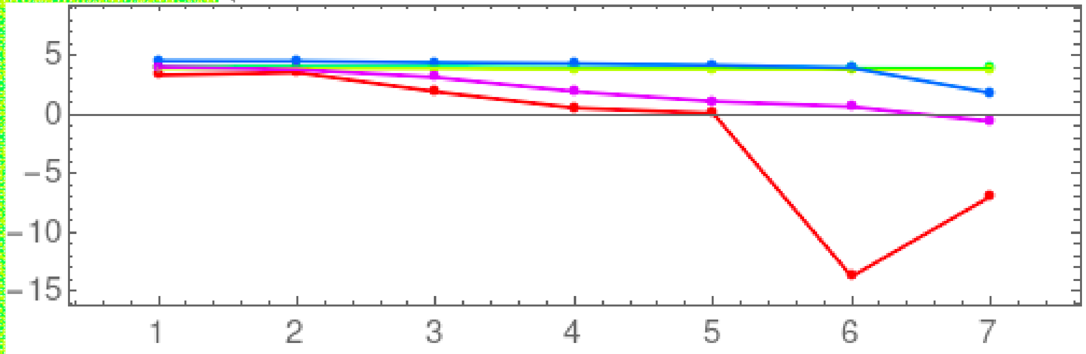




- JarKus LiDAR data of increasing quality from 2016 until 2022 (7 epochs)
- Selected terrain points (from 2 available classes, terrain and non-terrain)
- 300 x 400 m
- Interpolated to 1m grid -> 120 000 grid cells -> 120 000 time series

Results, k-means, k=5, full grid





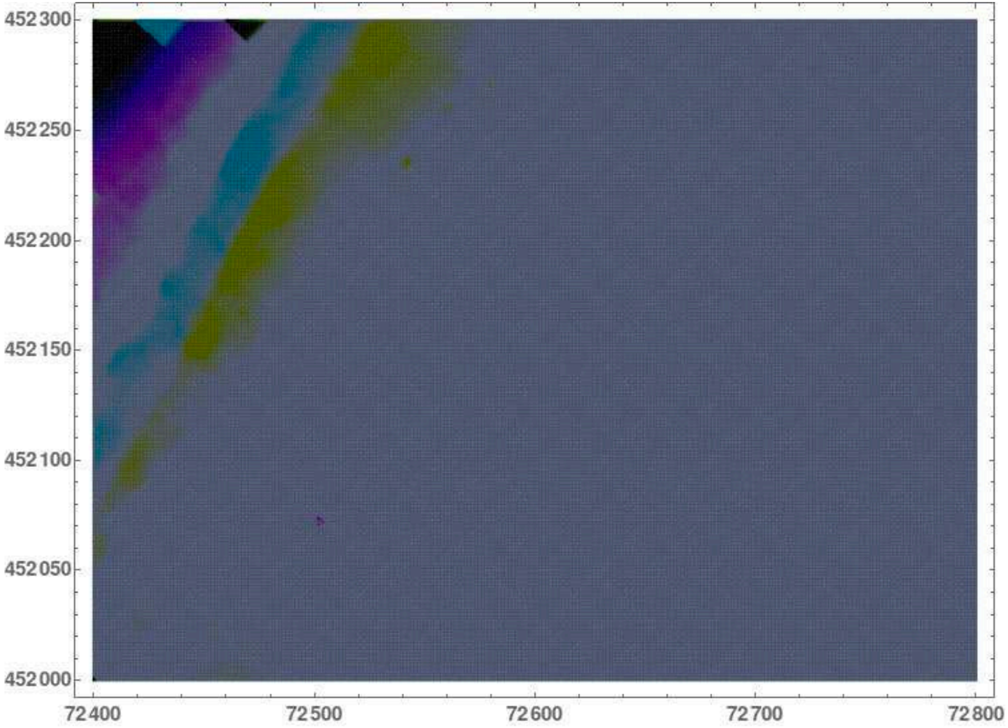
Left: grid cells colored by cluster;
Outlying time series present in red cluster

Right: average time series per cluster

Results, PCA, full grid

False color RGB showing loadings for first 3 principal components

Artefact at top left (two triangles).

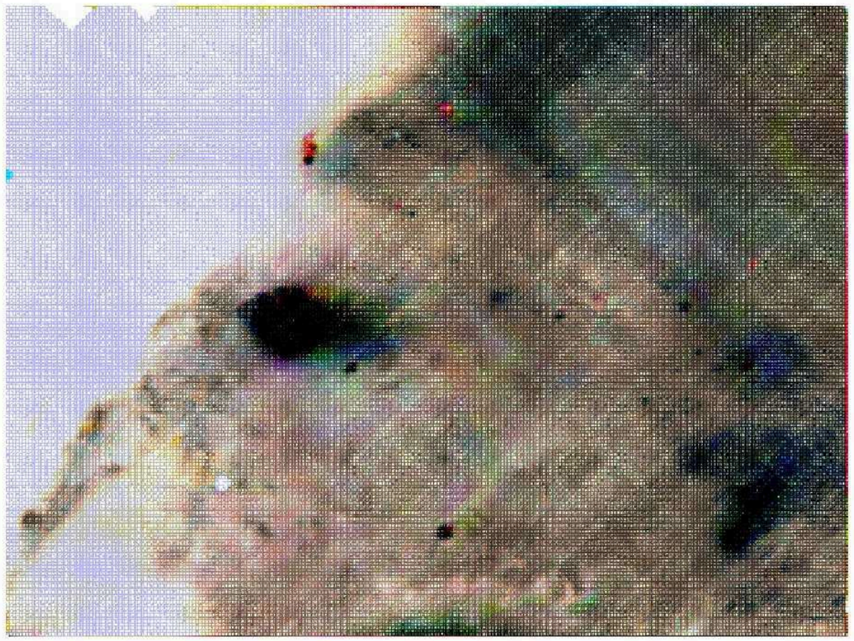


Idea:
remove first PCA cluster from data and repeat PCA analysis

Results, PCA after removal outlying cluster

False color RGB showing
loadings for first
3 principal components

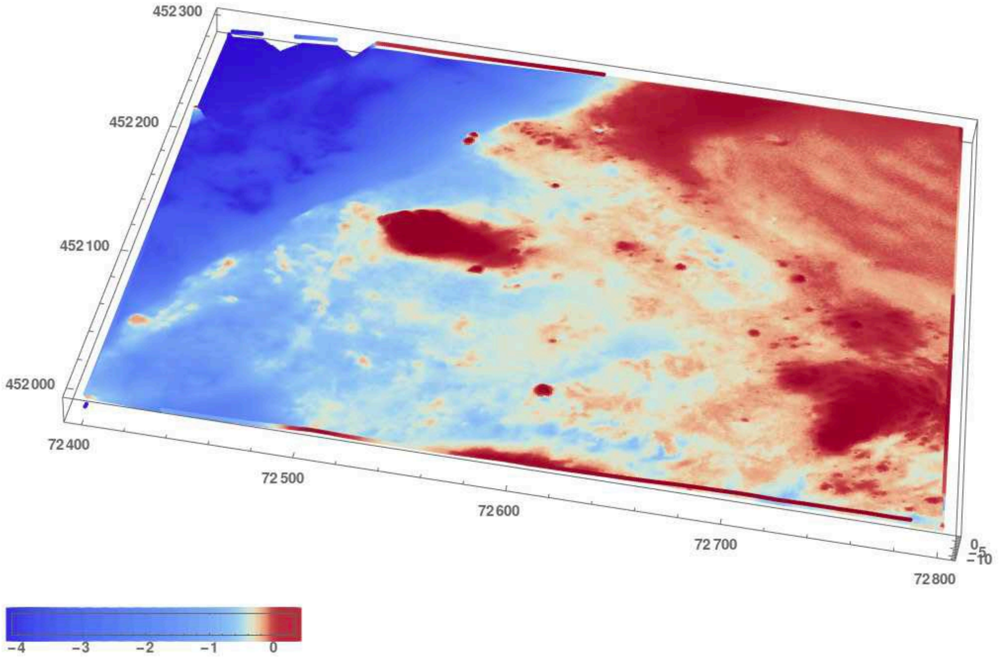
Artefact at top left removed



Results appear to highlight natural processes (to be further investigated)

PCA is sensitive to outliers (which is known)

Difference map, after removal outlying cluster



Difference between 2022 and 2016 grid after removal first k-means cluster

PCA highlights change regions, k-means ’doggingly’ separates change space

Conclusions on 4D analysis

4D data is more and more coming available

- Different repositories of Permanent Laser Scan data
- Repeated ‘institutional’ airborne laser scan and photogrammetric elevation data, and MBES data
- Repeated UAV and TLS campaigns

Multi-epoch analysis tools
- Only recently appeared
- Have been hardly applied on other projects than design-project
- Have been hardly combined

A Python Tool-box like https://github.com/3dgeo-heidelberg/py4dgeo allows to
- exploit different methods interactively
- find out peculiarities of methods on (certain) data
- also give non-experts access to upcoming methodology